INTERSPEECH 2017: Show & Tell Contribution
August 20–24, 2017, Stockholm, Sweden

Real-time Speech Enhancement with GCC-NMF: Demonstration on the Raspberry Pi and NVIDIA Jetson

Sean UN Wood, Jean Rouat

NECOTIS, Department of Electrical and Computer Engineering
Université de Sherbrooke, Québec, Canada
sean.wood@usherbrooke.ca, jean.rouat@usherbrooke.ca

Abstract

We demonstrate a real-time, open source implementation of the online GCC-NMF stereo speech enhancement algorithm. While the system runs on a variety of operating systems and hardware platforms, we highlight its potential for real-world mobile use by presenting it on two embedded systems: the Raspberry Pi 3 and the NVIDIA Jetson TX1. The effect of various algorithm parameters on subjective enhancement quality may be explored interactively via a graphical user interface, with the results heard in real-time. The trade-off between interference suppression and target fidelity is controlled by manipulating the parameters of the coefficient masking function. Increasing the pre-learned dictionary size improves overall speech enhancement quality at increased computational cost. We show that real-time GCC-NMF has potential for real-world application, remaining purely unsupervised and retaining the simplicity and flexibility of offline GCC-NMF.

Index Terms: Real-time, Speech Enhancement, Embedded Systems, GCC, NMF, GCC-NMF

1. Introduction

Various real-world speech processing applications depend on real-time speech enhancement algorithms. Despite this, many recently-developed algorithms are unsuitable for real-time use due to batch processing or computational requirements. We previously presented the offline GCC-NMF source separation algorithm and studied its performance on offline speech separation and enhancement tasks [1]. We have since developed an online GCC-NMF variant, and applied it to real-time speech enhancement [2]. In this demonstration, we present the real-time GCC-NMF speech enhancement algorithm running on embedded hardware platforms. A graphical user interface provides real-time visualization of the enhancement process and allows interactive manipulation of system parameters, such that the effects on subjective enhancement quality may be explored, with the results heard in real-time.

2. Real-time GCC-NMF

The GCC-NMF stereo speech enhancement algorithm [1] combines non-negative matrix factorization (NMF) dictionary learning [3] with the generalized cross correlation (GCC) localization method [4]. In the offline setting, the NMF dictionary is learned directly from the mixture signal (10 seconds in duration in our experiments), with enhancement then performed independently for each time frame. In the online setting, the NMF dictionary is pre-learned in an unsupervised fashion on a different dataset than seen at runtime, generalizing to new speakers, acoustic and noise environments, and recording setups. For each input frame, an angular spectrum is computed for each

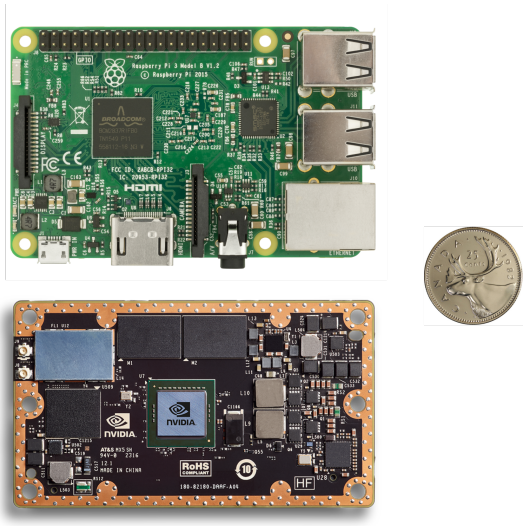

Figure 1: *Demonstration hardware platforms: Raspberry Pi 3 (above) and NVIDIA Jetson TX1 (below), with a quarter for scale.*

NMF dictionary atom d via the GCC-NMF function. A coefficient mask is then generated based on the distances between the target TDOA and the angular spectrum peak TDOAs, see (1).

2.1. User interface

Figure 2 depicts the graphical user interface for the GCC-NMF speech enhancement system, allowing visualization of the enhancement process and interactive modification of system parameters. Users may adjust the NMF dictionary size, the number of NMF coefficient updates at each time frame, and the parameters of the coefficient masking function M_d,

$$M_d = \frac{\exp\left(-\left(\frac{\left|\text{argmax}_\tau\left(\text{G}_{d\tau}^{\text{NMF}}\right)-\tau^*\right|}{\alpha}\right)^\beta\right)}{1+\eta} + \eta \qquad (1)$$

where d is the atom index, τ^* is the target TDOA (*Center*), α controls the window width (*Width*), β controls the shape of the window function (*Shape*), and η defines the minimum value of the mask (*Floor*), reducing perceptual artifacts.

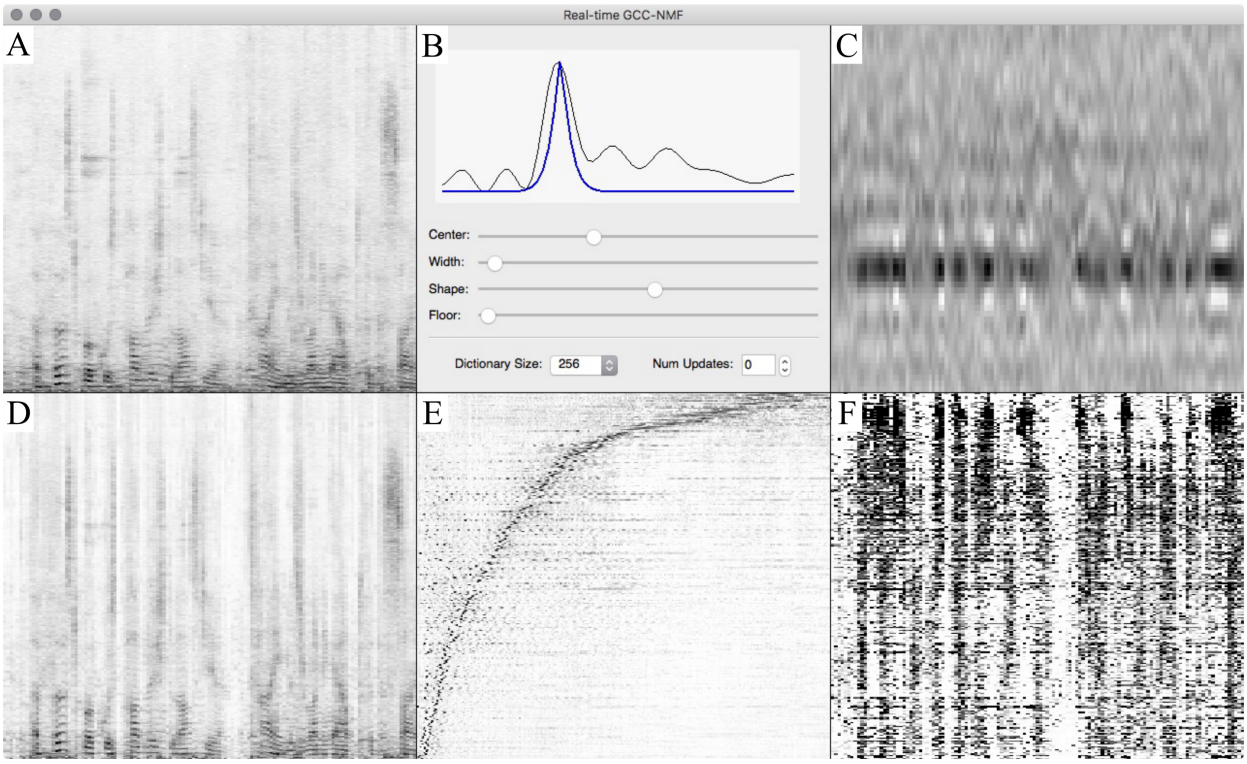

Figure 2: *Graphical user interface for the real-time GCC-NMF speech enhancement system. Panel B offers control of the coefficient masking function, depicted in blue with the GCC-PHAT angular spectrum in gray, as well as the NMF dictionary size, and number of NMF coefficient updates performed for each input frame. The other panels visualize elements of the enhancement system: the input spectrogram (A) and output spectrogram (D), as frequency versus time; the GCC-PHAT angular spectrogram (C) as TDOA versus time; the NMF dictionary (E), as atom index versus frequency (ordered by increasing spectral centroid); and the NMF coefficient mask (F), as atom index versus time. Panels are updated in real-time in waterfall plot style.*

2.2. Implementation details

The demonstration system is implemented in Python, using *Theano*[1] for GPU acceleration, and *pyqtgraph*[2] and *PyQt*[3] for the user interface. The audio sample rate is 16 kHz, with the STFT window size equal to 1024 samples (64ms) with a 512-sample hop size (32ms). Demonstration examples are taken from the SiSEC challenge speech in noise dataset [5], and the NMF dictionary is pre-trained on isolated speech and noise signals from the CHiME 2016 corpus [6]. We present the system on the NVIDIA Jetson TX1 GPU development board and the low-cost Raspberry Pi 3, though the software is cross-platform by design. Source code will be made available at `https://github.com/seanwood/gcc-nmf`.

3. Conclusion

We have presented an interactive demonstration of the real-time GCC-NMF stereo speech enhancement algorithm. The effect of various system parameters on subjective enhancement quality can be explored interactively by manipulating the parameters of the coefficient masking function, the dictionary size, and number of coefficient updates, offering control over the trade-off between interference suppression and target fidelity, as well as overall enhancement quality.

[1] http://deeplearning.net/software/theano

[2] http://www.pyqtgraph.org

[3] https://www.riverbankcomputing.com/software/pyqt/intro

4. Acknowledgements

The authors would like to thank NSERC and FRQNT/CHIST-ERA IGLU for funding, as well as the developers of the open source libraries Theano, pyqtgraph, and PyQt.

5. References

[1] S. U. N. Wood, J. Rouat, S. Dupont, and G. Pironkov, "Blind speech separation and enhancement with GCC-NMF," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 745–755, 2017.

[2] S. U. N. Wood and J. Rouat, "Real-time speech enhancement with GCC-NMF," in *Interspeech 2017*, 2017.

[3] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[4] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320–327, 1976.

[5] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2017, pp. 323–332.

[6] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.